PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/28, 17/30	A1	(11) International Publication Number: **WO 99/19814**
		(43) International Publication Date: 22 April 1999 (22.04.99)

(21) International Application Number: PCT/AU98/00849

(22) International Filing Date: 13 October 1998 (13.10.98)

(30) Priority Data:
PO 9779 14 October 1997 (14.10.97) AU

(71)(72) Applicant and Inventor: POUFLIS, Jason [AU/AU]; 12 Karara Street, The Gap, QLD 4061 (AU).

(74) Agent: GRIFFITH HACK; G.P.O. Box 4164, Sydney, NSW 2001 (AU).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: THE UTILISATION OF MULTI–LINGUAL NAMES ON THE INTERNET

(57) Abstract

A method for providing for multi–lingual names for use on the Internet, related networks, and computers is disclosed, the method comprising the steps of: forming an initial multi–lingual name in a multi–lingual format; mapping the multi–lingual name to a corresponding coded name in a reversible manner, the coded name comprising a restricted subset of the ASCII character set; utilising the corresponding coded name on the Internet, related networks and computers in place of the multi–lingual name. Preferably the mapping step further comprises adding a predetermined pseudo–root name server to the corresponding coded name. The mapping can include converting the multi–lingual name to a corresponding hexadecimal name and representing the hexadecimal name in an ASCII form. The corresponding coded name can be divided into a series of labels with each label having a predetermined portion comprising a control code for the label. The preferred embodiment is ideally utilised in existing or future Internet applications, utilities, resources or services. Existing uses include, but are not limited to: web browsers, editors, e–mail, ness, telnet, ftp, gopher, WAIS, whois, nslookup, trace, ping, finger, rpc, cgi programs, usernames, and databases. When performing queries the name server may respond with additional records for binary or sub–ASCII forms that match, or are variations of, the queried name. For example, if there are minor spelling errors, if they differ only in case, or their base equivalent characters are the same.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

The Utilisation of Multi-Lingual Names on the Internet

Field of the Invention

The present invention relates to the utilisation of multilingual names on the Internet, related networks and computer systems. Multilingual names include domain names, user names, file names, email addresses, newsgroups and Universal Resource Locators (URLs).

Background of the Invention

In recent times, the internet has undergone an explosive growth in utilisation. The original formation of the internet was based around the utilisation of English language character formats and as such, such formats dominate domain name structures, URLs etc. A large proportion of the world's population does not utilise the English language as its primary language of communication. Hence, there is a general need for other language's character based formats, for example: Chinese, Arabic, etc. Unfortunately, due to backward compatibility problems, these other language formats have received only restricted utilisation on the Internet. It is desired to expand the use of other languages to fundamental components of the internet being domain names, user names, file names, email addresses, newsgroups and Universal Resource Locators (URLs).

A glossary is provided, along with a brief Introduction to the Domain Name System (DNS), and references to the most relevant Request for Comments (RFCs).

Summary of the Invention

It is an object of the present invention to provide for an extended use of multilingual names on the internet, related networks and computer systems.

In accordance with a first aspect of the present invention, there is provided a method for providing for multilingual names for utilisation on the Internet, the method comprising the steps of: forming an initial

multilingual name in a multilingual format; mapping the multilingual name to a corresponding coded name in a reversible manner, the coded name comprising a restricted subset of the ASCII character set; and utilising the corresponding coded name (on the Internet) in place of the multilingual name.

Preferably the mapping step further comprises adding a predetermined pseudo-root name server to the corresponding coded name, particularly when the name is a domain name, or email address. The mapping can include converting the multilingual name to a corresponding Hexadecimal coded name and representing the Hexadecimal coded name in an ASCII form. The corresponding coded name can be divided into a series of labels with each label having a predetermined portion comprising a control code for the label.

The preferred embodiment is ideally utilised in existing or future internet applications, utilities, resources or services. Existing applications include, but are not limited to: web browsers, editors, e.mail, news, telnet, ftp, gopher, WAIS, whois, nslookup, trace, ping, finger, rpc, cgi programs, file names, usernames, and databases.

Brief Description of the Drawings

Notwithstanding any other forms which may fall within the scope of the present invention, preferred forms of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig. 1 illustrates the steps in the method of the preferred embodiment.

Description of Preferred and Other Embodiments

The preferred embodiment discloses processes that allow:

1. Multilingual names to be represented in limited subsets of the ASCII character set,

2. Names which are compatible with existing software applications and databases, thus requiring no change to

existing software.

3. New Software (and changes to existing software) to be made that incorporate the processes described, which may replace, or work with existing software.

Using the processes described, multilingual domain names can be utilised, without changes to existing resolver or name server software.

The preferred embodiment is fully backwards compatible with existing systems and does not require any changes to existing software used for processing domain names, user names, file names, email addresses, newsgroups and Universal Resource Locators (URLs).

Existing programs don't need to be changed, however it is expected they will progressively be adapted to make it easy for non-English alphabets to be read and typed in the form of domain names, email addresses, etc.

The preferred embodiment allows multilingual names to be written in many languages, even a mix, and then converted to fit into a subset of ASCII characters. A converting program is needed to do the conversion and display of Multilingual names.

By way of definition any program that converts between representations of names (multilingual name ↔ coded name) is called a converter - this may include resolvers, name servers, web browsers, and any program that carries out the converting process.

The preferred embodiment proposes, and address the issues of

1. General Methods that allow a variety and mix of representations of multilingual names;

2. Substitution of Characters for special words, or base equivalent characters;

3. Control Codes that indicate the encoding used, and splitting of names that are too long;

a. UCS-2 as Hex in ASCII which is a particular encoding and splitting method;

4. Pseudo-Root Names attached to hierarchical names, to indicate an alternative hierarchy;

5. Application to Names of particular types: strings, newsgroups, domain names, email addresses, and URLs.

6. Forms of Implementation covering software and interfaces.

Conventions

The following conventions are used in the following examples.

<> ASCII characters are in angle brackets eg. <Jason>

[ ] UCS-2 characters are in square brackets eg. [Jason]

Names with components or a hierarchy have usually been written with separators between the components such as the at symbol '@', dot '.', or slash '/'.

eg. news: "comp.law.patents";

email: "Jason@OneAccount.net";

URL: "http://www.OneAccount.net/login.cgi".

Since this invention allows these symbols to be used within components, these symbols only act as separators outside of brackets.

eg. "<Jason>@<OneAccount>.<net>";

"<http:>//<www>.<OneAccount>.<net>/<login.cgi>".

General Methods

A multilingual name may be a simple string, or may comprise a number of components that require parsing and interpretation, as part of conversion to a coded name. Components of names may be hierarchically organised from left to right or right to left and may contain other non-hierarchical components.

Implementors of converters have the choice of converting the entire string, or converting each component, since they are likely to be specialists in their target language market.

Converting is at least the reversible transformation of characters from a multilingual set to ASCII, and may

comprise parsing of components, substitution of characters, encoding, splitting, control codes indicating the encoding or splitting, or attachment of pseudo-root names.

Parsing of multilingual components involves identification of separators. Each separator can now be represented by several characters from several languages. The user may even be given the option of what symbols they would like to use as separator characters.

eg. instead of "@", it is possible to choose " at ", so that a corresponding example email address would be "Jason at OneAccount.net".

Substitution of Characters

Special Words

Parts of a multilingual name may have special meaning, for instance, the file name extension, or protocol to use. A Japanese language user may prefer to see and use the Japanese characters for ".exe", or "http:".

Converters may substitute ASCII characters in place of the synonymous multilingual characters.

Base Equivalent Characters

Sometimes, it is desirable to ignore the case of characters in English, such as for searching or matching names. We call this being case insensitive. To make comparisons, it is usual to force all the characters to upper or lower case. Other languages' alphabets have different rules. For instance, Greek has three forms of Sigma, one only for use at the end of a word, when the word is lowercase.

Different kinds of comparisons may be done for each alphabet. We therefore define a sets of characters that are equivalent to each other for purposes of comparison. From each set, one character is said to be the Base Equivalent Character. When making that comparison, equivalent characters are forced to the base equivalent character.

For Case Insensitive comparisons on UCS-2, it is preferred that the base character be the earliest character

of each set in ISO10646 order, from within the language. This forces Latin, Greek and Cyrillic to uppercase and Hiragana and Katakana to lowercase. So, for instance, Greek lowercase alpha is substituted with Greek uppercase alpha, but not with Latin "A", nor Cyrillic "aleph".

Another type of comparison could be character shape. The letters "IBM" could be Latin, Greek or Cyrillic. A language insensitive search could force them all to Latin.

Control Codes

Control codes can be attached to a coded name, or to each component of a coded name, to indicate the type of encoding, and the split sequence. A particular example is UCS-2 as Hex in ASCII.

Method of Encoding

When a multilingual name is converted into a coded name, control codes can be attached to the coded name to indicate the method of encoding.

Split Sequence

If a component of a multilingual name is too long when converted to fit into a single component of a coded name, it may be split across several components of a coded name. Control codes attached to each component of the coded name can indicate which part of a multilingual component it belongs to, ie its order in a split component.

This is particularly useful for hierarchical names with limits on the length of components, such as domain names.

UCS-2 as Hex in ASCII

UCS-2 as Hex in ASCII is an encoding of multilingual names. Its 3 octet control code is <X-n> where n is an ASCII number from <1> to <9>, when it comprises a split component, and <0> when the component is not split. The control code is prepended to the coded component.

Each UCS-2 character becomes four ASCII characters in the ranges <0>-<9>, <A>-<F>; representing the value of the UCS-2 character in Hexadecimal.

An example of UCS-2 to ASCII, not split.

[Jason] -> <X-0004A00610073006F006E>

An example of split ASCII to UCS-2

<X-30065><X-2006E><X-1004F> -> [One]

Pseudo-Root Names

A pseudo-root name is a predetermined name attached to coded hierarchical names, such as newsgroups and domain names, so that they become part of a predetermined hierarchy. By adding the pseudo-root name to all coded names, that branch of the hierarchy effectively becomes the root of a pseudo-hierarchy.

This has several useful properties:

1. Separation of Names

Coded names won't be mixed up with normal ASCII names, so it is less confusing for users.

2. Separation of Risk

Technical, business or political changes to the pseudo-root hierarchy names, won't adversely affect the real root or other branches.

3. Separation of processing load

In hierarchical distributed systems, such as DNS, the processing load arising from multilingual names, is allocated to computers serving the pseudo-hierarchy.

4. Specialisation

Pseudo-root hierarchies can specialise in a particular type of encoding or language. Different converters can attach different pseudo-root names, meaning the converter programs and hierarchies can specialise.

5. Politics

A pseudo-root can be made in a part of the hierarchy in which control is exercised.

It is recommended that all coded domain names are subdomains of "X-X.NET", and coded newsgroups created under "alt.x-".

Application to Names

Many combinations of processes may be applied to various kinds of names:

Strings

Simple multilingual strings, such as user names, might merely be converted to a coded form with a control code attached indicating the encoding method, such as X-0.

Strings with components, such as file names, might also have special words substituted with synonymous characters. For instance, a Japanese file name is suffixed by Japanese characters that indicate it is an executable program, these characters may be replaced by the file name extension ".exe".

Newsgroups

Newsgroups are also known as Internet News, and Usenet.

Coded names can be used as the names of newsgroups, and displayed to users as multilingual newsgroup names.

To name newsgroups in multilingual characters, with an example for a newsgroup about patent law in English.

[Law, Patent] (English language)

1. Substitute with base equivalent characters. Substitute ISO language code for language. <EN>.[LAW].[PATENT]

2. Convert UCS-2 to ASCII and add control codes. <EN>.<X-0004C00410057>.<X-00050004100540045004E0054>

3. Add pseudo-root for multilingual news hierarchy. <ALT>.<X->.<EN>.<X-0004C00410057>. <X-00050004100540045004E0054>

4. Present the normal ASCII name of the newsgroup. "ALT.X-.EN.X-0004C00410057.X-00050004100540045004E0054"

It is recommended that since some alphabets are shared by many languages, that the top level newsgroup names be the 2 letter ISO language codes.

Domain Names

A brief introduction to the domain name system is supplied later. For details see the referenced RFCs.

Domain Names are hierarchical names commonly used to identify organisations on the internet. RFC1035 specifies the presentation of domain names as domain labels separated by '.' dots, with the highest level domain label on the right, and subdomains proceeding to the left. For example in "www.example.com.au.", "au" is the top level label for Australia, "com" is the second level label for commercial enterprises, "example" is the third level label - the name of the enterprise, and "www" is the fourth level label identifying a computer in the enterprise. This is the traditional way of writing domain names.

Instead, the presentation of domain names is left to implementors of converters. The implementors, or even the users, may select appropriate separator, quote, and escape symbols, along with special words, and the direction of the hierarchy (left to right, right to left, etc.). Each domain label could even be entered in separate text fields, eliminating the need for separate characters. However, it is often easier to write and type a domain name with separating characters.

The domain names system is concerned with the format of binary data between resolvers and name servers. Due to compatibility issues, only a limited subset of ASCII is used in labels, the characters 'A'-'Z', 'a'-'z', '0'-'9' and '-'. It is an object of the preferred embodiment to allow multilingual domain names to be represented in this subset of ASCII.

A process for representing multilingual domain names can be shown in Fig. 1.

1. Parsing, and Substitution of Special Words 1;
2. Substitution of Base Equivalent Characters 2;
3. Encoding, Splitting and Control codes 3;
4. Adding pseudo-root domain name 4;
5. Presenting coded form of names 5;

1. <u>Parsing, and Substitution of Special Words</u>

Converters may accept domain name labels in a variety of ways, such as selection from a list of countries, or typing a partial domain name into a text field. Converters which allow labels to be typed together into one field need to parse the parts of the domain name into labels. Separator, quote, and escape characters may be defined by implementors of the converter, or be left to the user's choice.

Special words may be substituted for selected or typed labels. For instance, replacing the Arabic label for Australia with "au", or the Thai label for business with "com".

2. Substitution of Base Equivalent Characters

English domain names are case insensitive, so lowercase Latin should be replaced with uppercase. Other languages may have different preferences. Defining the sets of equivalent characters can be left to implementors, and specialists in that language.

3. Encoding, Splitting and Control codes

The Internet standard RFC1035 specifies that domain names have an overall limit of 255 octets, and that each label has a limit of 63 octets. Currently, labels only contain ASCII characters 'A'-'Z', 'a'-'z', '0'-'9' and '-'.

It is possible in future that labels could be made of 8bit (ASCII, ISO8859), 16bit (UCS-2), 32bit (UCS-4), or variable length characters (UTF-8, UTF-7). Labels could even be made of other data, such as bitmaps (pictures), or sound data.

For the representation of multilingual domain names, the preferred method of encoding is UCS-2 to Hex in ASCII, as it is fully compatible with existing DNS tools.

Since each UCS-2 character maps to 4 ASCII characters, any label that is longer than 15 UCS-2 characters must be split, so that it fits into the maximum label length of 63 octets. It is further recommended that labels which are 15 UCS-2 characters long, should be split with a coded

blank second part. This allows for separation of control of the common part of a shared domain label, as will be further explained below.

There may be several businesses that share the first part of their name. Rather than giving control of the common part to one of these businesses, it is possible to give control to a neutral third party, such as the superdomain.

For example:

[Traveller's Rescue].<AU>

[Traveller's Rest].<AU> ,

and

[Traveller's Res].<AU>

when split and prefixed would become

<X-2>[cue].<X-1>[Traveller's Res].<AU> ,

<X-2>[t].<X-1>[Traveller's Res].<AU>,

and

<X-2>[].<X-1>[Traveller's Res].<AU>

Control of the common domain <X-1>[Traveller's Res].<AU> could be given to <AU>, or shared by the organisations. Each organisation can have control over its <X-2> subdomain.

4. Adding pseudo-root domain name

A pseudo-root domain name is added to the coded domain name, for the reasons mentioned in "Pseudo-Root Names". Name servers for the pseudo-root may be specialised for the processing of names in a particular encoding, or language.

The recommended pseudo-root domain name to add is <X-X>.<NET>. That is, "X-X.NET." .

5. Presenting coded form of name

A converter may have to present the coded form in a way which is useable by applications. The traditional way is specified in RFC1035 - labels separated by dots, with the highest level label to the right.

Converters that query the DNS themselves, may not need to concatenate the labels into a contiguous string.

Example of converting Multilingual Domain Name

The following provides an example of the domain name conversion process of the preferred embodiment.

"Glebe, Traveller's Rescue, Australia"

1. Parsing, and Substitution of Special Words

-> [Glebe].[Traveller's Rescue].<AU>

2. Substitution of Base Equivalent Characters

-> [GLEBE].[TRAVELLER'S RESCUE].<AU>

3. Encoding, Splitting and Control codes

Encoding UCS-2 characters as Hex in ASCII

-><0047004c004500420045>.<0054005200410056004500 4C004C004500520027005300200052004500530043005500 45>.<AU>

Splitting and Prefixing with Control codes

-><X-00047004c004500420045>.<X-2004300550045>.<X-10054005200410056004500 4C004C004500520027005300200052004500 53>.<AU>

4. Adding pseudo-root domain name

-><X-00047004c004500420045>.<X-2004300550045>.<X-10054005200410056004500 4C004C004500520027005300200052004500 53>.<AU>.<X-X>.<NET>.

5. Presenting coded form of name

->X-00047004c004500420045.X-2004300550045.X-1005400520041005600450 04C004C004500520027005300200052004500 53.AU.X-X.NET.

Email

Email mailboxes and addresses can use a larger part of the ASCII character set than DNS. Normally, an email address comprises a mailbox name (local part) at a domain name.

A multilingual email address could be formed in some other way, using the languages own symbols for addressing. For instance, [Jason at Home, Australia] instead of Jason@HOME.AU. Converters or mail programs are responsible for processing the email addresses correctly. Multilingual addresses could be processed in a number of ways:

1. Parsed, coded and sent to a mailbox at a domain

Parsed

->[Jason]@[Home].<AU>

Coded

-> <X-0>[Jason]@<X-0>[HOME].<AU>.<X-X>.<NET>.

2. Coded, and sent to a converting mail exchanger

-> <X-0>[Jason at Home, Australia]@<MAIL>.<X-X>.<NET>.

3. Coded, and resolved by DNS

-> <X-0>[Jason at Home, Australia].<MAIL>.<X-X>.<NET>.

4. Parsed, coded, and resolved by DNS

-> <X-0>[Jason at Home].<AU>.<MAIL>.<X-X>.<NET>.

Universal Resource Locators (URLs)

URLs encompass file names, newsgroups, domain names, email, and many other names. A larger part of the ASCII character set is available for names, and encoding of octets is provided for. However, the schemes that URLs encompass remain restricted in the characters they can use, so there is a need for coded multilingual URLs.

Substitution of special words and symbols

URLs are currently defined for the US-ASCII character set. Multilingual users may prefer to use symbols from their own language, in place of the specific scheme names, reserved and special characters. Converters would then parse these symbols and replace them with the US-ASCII symbols.

For instance [Secure Web] -> <https:> or [web] -> <http:>.

Schemes that use Internet protocols, are formatted as: "<scheme>://<user>:<password>@<host>:<port>/<url-path>". Multilingual scheme should be parsed into a coded form like this. Conversion of components using the UCS-2 as Hex in ASCII can be applied to the user name, password, and host name (which is a usually a domain name), and components of the url-path.

Multilingual port numbers should be converted into synonymous ASCII number, if written as a non-ASCII number such as in Chinese or Sanskrit numerals.

The url-path may be further parsed, and broken down into special and reserved characters, path names, file names, search, argument names, and argument values.

It is left to implementors of converters to elect the characters and symbols in their language, that will substitute for scheme names, special and reserved characters.

Some Examples - parsed and substituted, but not coded.

[Mail: Jason at Home, Australia]

-> <mailto://[Jason]@[Home].AU.X-X.NET

[News: English, Patent Law]

-> <news://alt.x-.en.[law].[patent]>

[Secure Web: OneAccount - login(Jason)]

->http://[OneAccount].X-X.NET./[login].cgi?[login]=[Jason]>

[Local File: Patents - Multilingual Test, program]

-><file://localhost/[Patents]/[Multilingual Test].exe>

Forms of Implementation

The method of the preferred embodiment can take many different forms of implementation, for example, as follows:

Stand Alone Converter

This form takes in a multilingual name, and outputs a coded name as an ASCII string, or some other representation. The converter may be created to work for particular kinds of names, such as URLs or email addresses, and/or to work with particular applications, such as web browsers.

Converters may have controls to, or automatically, send the ASCII string to relevant applications. They may allow a user to copy and paste to and from their applications.

Incorporated into applications

Alternatively, the conversion function may be incorporated into the applications such as browsers, editors, email, telnet, ftp, and news.

Plug-in or add-on to application

The converter may be a program or library that plugs-in or adds onto the existing applications, providing the application with the added multilingual name functionality.

Application loadable control

The converter may take the form of a control that the application can use. Examples are Web pages that include javascript, java controls, or Active-X controls.

Such controls and plug-ins may replace, or overlay a browsers current URL entry field, with a multilingual name field. This field both displays the multilingual name, and allows entry of multilingual URLs. Coded names are passed back and forth from converter to browser.

Web Page interfaces to converter

A converter may run on a web server, with access to the converter being provided through multilingual web pages. Users access a multilingual URL/domain name service such as "http://X-X.NET/". If their browser requests a particular language, a web page in that language is provided (if available), otherwise a multilingual page is provided.

The web page can typically provide a form, so that the user may type in a multilingual URL. Users may select common parts from lists such as the encoding scheme, organisation type, and country. These lists may have defaults on a per user, or per language basis.

When the multilingual URL form is submitted, the converter server has several options:

1. returning the coded URL as an ASCII string, which the user may link to, or use as they please.
2. providing a redirection to the coded URL.
3. presenting a frame view, where one frame contains the requested coded URL, and another contains a multilingual URL form, for typing other URLs.

Multilingual Registries may also provide a web interface to provide for registration of multilingual

names, such as domain names and email addresses.

Converter packaged with other facilities

Converters may be packaged with other facilities. For instance, a program may parse a multilingual name in several ways, and perform several searches such as DNS lookup, whois search, and web page search. It might present information to a user, or return specific information to a client application.

Resolvers

The resolver accepts the multilingual name direct from applications, but then converts it before querying name servers. Resolvers may query name servers for both the binary and sub-ASCII representations of the multilingual domain name. The resolver may also try variations on the name.

Name Servers

When performing recursive queries, the name server accepts sub-ASCII or binary multilingual domain names; and queries other name servers with sub-ASCII or binary Multilingual domain names.

The name server may convert from binary name to another format before querying its database and may return answers for either form.

In responses, the name server may respond with additional records for binary or sub-ASCII forms (including CNAME and A records) that match, or are variations of, the queried name. For example, if there are minor spelling errors, if they differ only in case, or their base equivalent characters are the same.

Databases

Databases may keep records in binary or sub-ASCII form. Conversion between them, and conversion for client or server programs may be required.

Other areas of application

The principle of having the first 3 characters in a

field represent the encoding scheme can be applied generally. This can be applied to directory services, such as Whois, LDAP, and to search engines, and to databases.

It can therefore be generally seen that the preferred embodiment provides for the representation of Multilingual characters, in more limited character sets. In particular, the process includes converting UCS2 to Hex in ASCII, applied to internet names used in the Domain Name System (DNS), email, news and Uniform Resource Locaters. For DNS, a multilingual domain label is represented in one or more sub-ASCII labels. The first 3 characters identify the label's encoding scheme, leaving a maximum of 60 sub-ASCII characters for encoded data in each domain name label.

In UCS-2 to Hex in ASCII encoding the first and second characters is the name of the scheme 'X-'; and the third character identifies the part of the split multilingual label. The name of the pseudo root server "X-X.NET" is attached to the sub-ASCII representation of the multilingual domain name. The pseudo root server is visible in the current domain name space. For email, the first three characters of the local-part identify the local-part's encoding scheme. The domain name follows the rules for DNS.

Alternatively, the entire email address is encoded, and sent to the relevant mail server, exchanger or gateway for processing or forwarding. For URLs, the first three characters of each component (name, label, argument) in the URL identifies the encoding scheme.

The encoding and representation can be implemented in the form of various software devices, such as upgrades or add ons to existing software, incorporation in new software, stand-alone applications, databases, servers, clients, resolvers, name servers.

The first three characters identify the encoding scheme to a converter, so that it may display the name in the right character set. These characters mean nothing to

existing DNS, E.mail and web systems, simply identifying the name of a domain, mailbox, file or other data. Hence variations utilising different encoding identifiers can also be easily used.

This scheme can be designed for temporary use, up until applications and databases, (including name servers and resolvers) become compliant with a multilingual character set such as ISO10646 or Unicode.

It is further possible under this scheme to have several pseudo roots. This allows multiple registries to run, specialising in particular languages. However, It is recommended that one pseudo root be selected, with registries sharing the pseudo root's database.

It would be further appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The described present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

Glossary

The following terms are hereinafter defined for ease of understanding:

Multilingual Name - made of non-ASCII characters, may be a string of characters, or several labels or fields. This specifically includes, and is not limited to, domain names, user names, file names, email addresses, newsgroups and Universal Resource Locators (URLs).

Coded Name - a string, or fields, of ASCII characters that represent a Multilingual name in some encoding.

Converter - any program that converts from one representation of names to another. Especially, converting from UCS-2 to Hex in ASCII and back. Converters may incorporate resolvers, and other functions such as substitution for equivalent characters.

ASCII - A character set that contains the English Alphabet, Arabic Numerals, punctuation marks and some computer control codes. There a several varieties of ASCII

Sub-ASCII - The limited subset of the ASCII character set that has been used in domain names: 'A'-'Z', 'a'-'z', '0'-'9', and '-' (dash).

UCS - Universal multi-byte Character Set encodings of ISO10646 and Unicode, which cover most living languages. UCS-2 is 2 bytes (16 octets), UCS-4 is 4 bytes.

Equivalent Characters - characters that are mapped to the same base character by a program. In English 'A' and 'a' differ only in case. To case insensitive programs, such as DNS, they are equivalent. In other languages, equivalent characters may differ in other ways. Eg. In Greek, there are two lowercase sigmas; one for use at the end of a word. Developers of programs for different language markets are specialists in these areas; they decide on which characters are equivalent.

Domain Name - a name upto 255 octets made of several labels, one for each level in the hierarchy. "www.x-x.net." is a domain in the "x-x.net." domain in the "net." domain. The DNS store information related to domain names.

Label - part of a domain name, upto 63 octets.

DNS - The Domain Name System. A distributed database that is accessed by resolvers asking name servers. The DNS stores computer's names, IP addresses, and more. See RFC 1034, 1035 and others.

IP address - A 4 byte internet network address.

Resolver - a program that applications use to query the DNS. A resolver in turn asks Name Servers for information.

Name Server - a name server has information about its domain that it gives to resolvers and other name servers. If it doesn't know it may query other name servers.

Root Name Servers - the name servers at the top of all hierarchies.

Pseudo-Root Name Servers - some application may add a predetermined name to all of their domain name queries, making it seem as if that name server is at the top of all hierarchies.

RFC - Request for Comments documents describe how the internet works. The Internet Engineering Task Force draws internet standards from the list of RFCs.

Introduction to the Domain Name System (DNS)

By way of introduction to the internet's Domain Name System, we illustrate with an example.

When a user wants to view a web page, they may type in or select it's URL. For example, a superannuation web page URL *is* "http://www.superannuation.net/index.htm". "www.superannuation.net" is a domain name, that is the name of the computer on which the page is kept. That computer's IP address (internet number) must be found to get the page. This is done by asking the DNS.

The web browser asks a DNS Resolver to find the IP address of the domain name. The Resolver asks the local name server for the address. If the local name server doesn't know, it then tracks down the address by asking other name servers. The local name server asks the net. domain name server where the superannuation.net. domain name server is. Then it asks this subdomain name server for the IP address of the domain name www.superannuation.net, which is 105.42.3.5 (just an example address).

The local name server then tells the resolver the IP address, which in turn informs the web browser. The web browser now asks the computer at that IP address for the web page via http: "//www.superannuation.net/index.htm". Internet Applications such as web browsers, ftp, telnet and email programs all use resolvers to ask the DNS for the address of domain names. Sensible domain names are easier for people to remember than IP addresses; when they are in their own language. To date, DNS implementations have required names to be in a small subset of ASCII: the

letters A-Z, digits 0-9, and the dash -.

Internet standard documents are readily available on the Internet. The most pertinent to this patent application is RFC1035 : Domain Names - Implementation and Specification which describes how DNS works, and the format of names in detail.

The DNS specification RFC1035, with further updates and clarifications, state that domain name labels may contain up to 63 octets of binary data. It is suggested that the names be made from the characters A-Z, 0-9 and - dash, a restricted subset of US ASCII, so that legacy applications keep working.

Until all internet applications and protocols (including resolvers, name servers, and databases) are able to handle binary labels, it is desirable to represent binary labels in this subset of ASCII, especially multilingual domain names.

Existing RFCs and Drafts

By way of background, a number of RFC documents, and internet-drafts are available from the Internet Engineering Task Force at http://ietf.org/".

http://dxcoms.cern.ch/wwwcs/public/ip/draftslist.html -

Although these documents frame the way in which the internet should work, a number of recommendations have not been adopted, nor implemented.

RFC882 Format of ARPA Internet Text Messages defines internet mail, and specifies the format of email addresses. RFC1035 Domain Names - Implementation and Specification defines the DNS protocol, and specifies a format for domain names as a sequence of labels separated by dots. Labels begin with a letter, and may contain characters from 'A'-'Z', 'a'-'z', '0'-'9' and '-' dash.

RFC1123 Requirement for Internet Hosts allows domain labels to begin with letters or numbers.

RFC1738 Uniform Resources Locators (URL) specifies the format of URLs, in a subset of US-ASCII that permits binary

data as octets represented by %HH, where H is 0-9,A-F more commonly known as 'Hex in ASCII'.

RFC2130 Character Set Workshop Report recommends ISO10646 as base character set for internet also says DNS should stay in limited ASCII format.

RFC2152 UTF-7 A mail safe transformation format for Unicode specifies methods for encoding Unicode into mail messages, but not for mail addresses, domain names, nor URLs.

RFC2181 Clarifications to the DNS Specification clarifies that 'any binary string whatever can be used as the label'.

RFC2070 Internationalisation of the Hypertext Markup Language is one of many RFCs, that describe multilingual documents, but do not address the issue of DNS, email or URLs.

RFC1468 for Japanese, RFC1557 for Korean, RFC1922 for Chinese specify encodings for these character sets, that begin with escape sequences.

It would be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

We Claim:

1. A method for providing for multilingual names for use on the Internet, related networks and computers, said method comprising the steps of:

forming an initial multilingual name in a multilingual format;

mapping said multilingual name to a corresponding coded name in a reversible manner, said coded name comprising a restricted subset of the ASCII character set;

utilising said corresponding coded name on the Internet and related networks in place of said multilingual name.

2. A method as claimed in claim 1 wherein said mapping step further comprises adding a predetermined pseudo-root name to said corresponding coded name.

3. A method as claimed in any preceding claim wherein said mapping includes converting said multilingual name to a corresponding hexadecimal coded name and representing said hexadecimal coded name in an ASCII form.

4. A method as claimed in any preceding claim wherein said corresponding coded name is divided into a series of labels with each label having a predetermined portion comprising a control code for said label.

5. A method as claimed in any preceding claim wherein a multilingual name is parsed or broken down into components.

6. A method as claimed in any preceding claim wherein components a multilingual name that have special, reserved, or schematic meaning are replaced with synonymous components in the coded name.

7. A method as claimed in any preceding claim wherein the characters of a multilingual name are replaced with their base equivalent characters.

8. A method for providing in domain name systems, answers that contain additional information about names that are similar to names in questions to the domain name

system, the names being multilingual, coded or ordinary ASCII.

9. A method as claimed in any preceding claim wherein said method is used in applications that also directly or indirectly use internet protocols or services.

10. A method as claimed in any preceding claim wherein said method is used in internet applications, utilities, resources or services.

11. A method as claimed in any preceding claim wherein a multilingual name is represented by a coded name for the purposes of sending, receiving or otherwise processing one of email, talk, chat, IRC, the coded name being a name for a user of a program, computer system, or network.

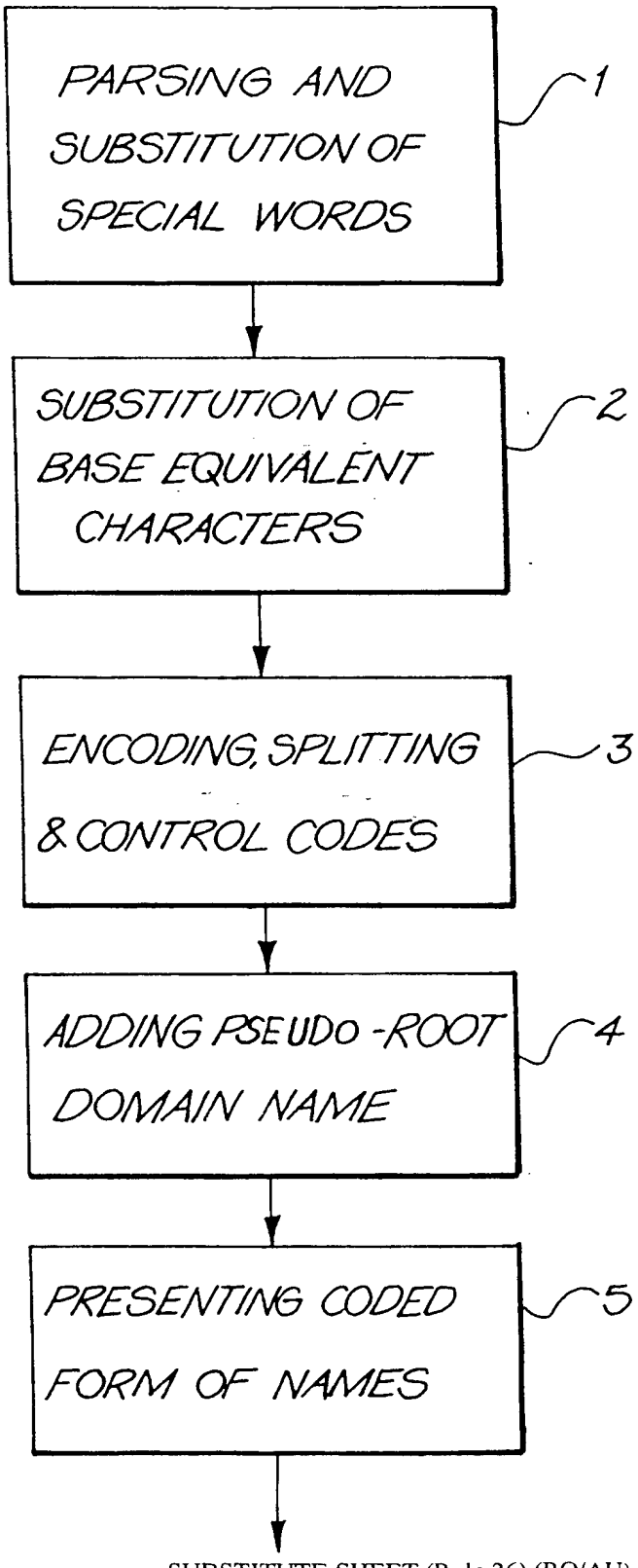

FIG. 1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU 98/00849

A. CLASSIFICATION OF SUBJECT MATTER

Int Cl[6]: G06F 17/28, 17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
AU: IPC as above, INTERNET

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
WPAT: INSPEC
JAPIO:

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	EP 0817099 (SUN MICROSYSTEMS, INC) 7 January 1998 whole document	8-10
X	US 5337233 (HOFERT et al) 9 August 1994 whole document	1-7,9-11
X	http://www.iahc.org/contrib/draft-duerst-dns-i18n-00.txt, "Internationalisation of Domain Names", M Duerst, University of Zurich, 10 December 1996, viewed on 9 November 1998.	1-7,9-11
A	US 5572668 (SEE et al) 5 November 1996 In particular column 5 lines 9-20	1-7,9-11

[X] Further documents are listed in the continuation of Box C

[X] See patent family annex

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 November 1998	20 NOV 1998
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200 WODEN ACT 2606 AUSTRALIA Facsimile No.: (02) 6285 3929	Authorized officer SEAN APPLEGATE Telephone No.: (02) 6283 2207

Form PCT/ISA/210 (second sheet) (July 1998) copbko

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU 98/00849

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5047932 (HSIEH) 10 September 1991 whole document	1-7,9-11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/AU 98/00849

Box 1 Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a)

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Claims 1-7: Coding a multilingual name into an ASCII subset.

Claim 8: Providing information on similar domain names.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims

2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/AU 98/00849

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member					
EP	817099	JP	10063597				
US	5337233	JP	6089304				
US	5572668	AU	41992/96	JP	2782478	WO	96/0815

END OF ANNEX